2D skeleton-based Human Action Recognition using Action-Snippet Representation and Deep Sequential Neural Network

Author: Aizada Askar
Supervisor: Nguyen Anh Tu
Co -supervisor: Min-Ho Lee

School of Engineering and Digital Sciences
Master of Science in Data Science

Outline

- Introduction
  - Background
  - Motivation
  - Problem
  - Objectives
- Related works
- Methodology
- Experiments and results
- Conclusion

Introduction

What is Human action recognition?

- recognizing the nature of an action
- typical activities performed indoors and outdoors
- recognized based on different sensors
- vision-based action recognition

Introduction

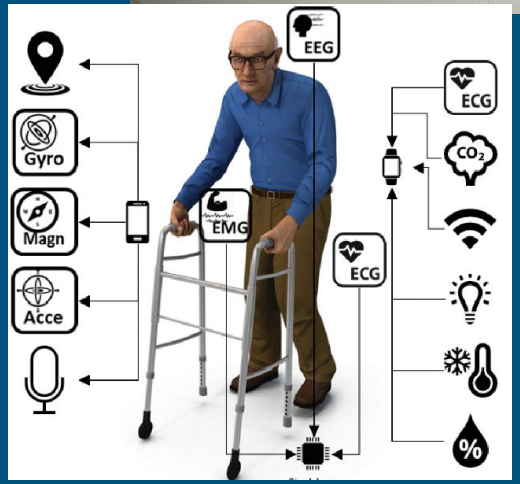
Why to study Human action recognition?

Important for:

- multimedia content analysis
- event interpretation
- behavior understanding

Central function of:

- intelligent surveillance
- smart healthcare
- crime



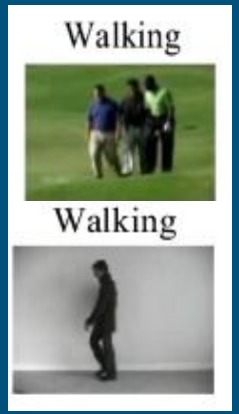
Introduction

Problem statement

- How to accurately and efficiently recognize human actions from video sequence

Challenges

- Anthropometric, multiview variation
- Cluttered and dynamic background
- Occlusion
- Insufficient data



Objectives

- **Different geometric feature extraction schemes**
  - capture the spatial and temporal relationship between poses
- **Effective classification models**
  - learn the deep correlations of long consecutive video sequence

Related works

Non-skeleton based approaches

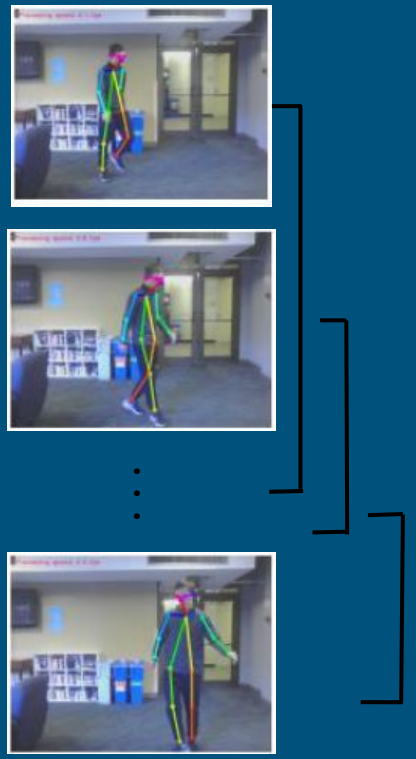
Method	Description	Limitations
G.Paoletti et al - SCAR[1]	finds subspaces of data points performs clustering better handles the temporal dimension of the data	• sensitive to the quality of foreground detection techniques • use human insights • limited to conventional machine learning tasks
L.Liu et al - LLC[2]	capture the correlations between similar descriptors	
C.Huang et al - HOF/HOG[3]	counts occurrences of gradient orientation recognition rate can be improved	
Mohana et al - STIP[4]	ignores the ST interrelationships between the all types of person visual features	

Related works

Skeleton based approaches

Method	Description	Limitations
HAR Using Depth Sensors[5]	• extract action templates from 3D data • 3D coordinates are preprocessed by rotation, translation and scaling techniques	• often noisy ➢ localizing body parts ➢ sensor range errors ➢ occlusions • have a minimal working range • poor performance in outdoor environment
HAR using pose kinetic energy[5]	• define key poses with normalized coordinates • computes atomic action templates	
Action-XPose[6]	• retrieves low and high level features from 2D skeleton data • fed into LSTM	• use frame-wise representation • do not consider geometric relationship between skeletons • difficulties in discriminating pose-similarity actions
Two branch stacked LSTM-RNN [7]	• based on 2D skeleton data • segmented into two parts	

Proposed solution



2D skeleton based approach

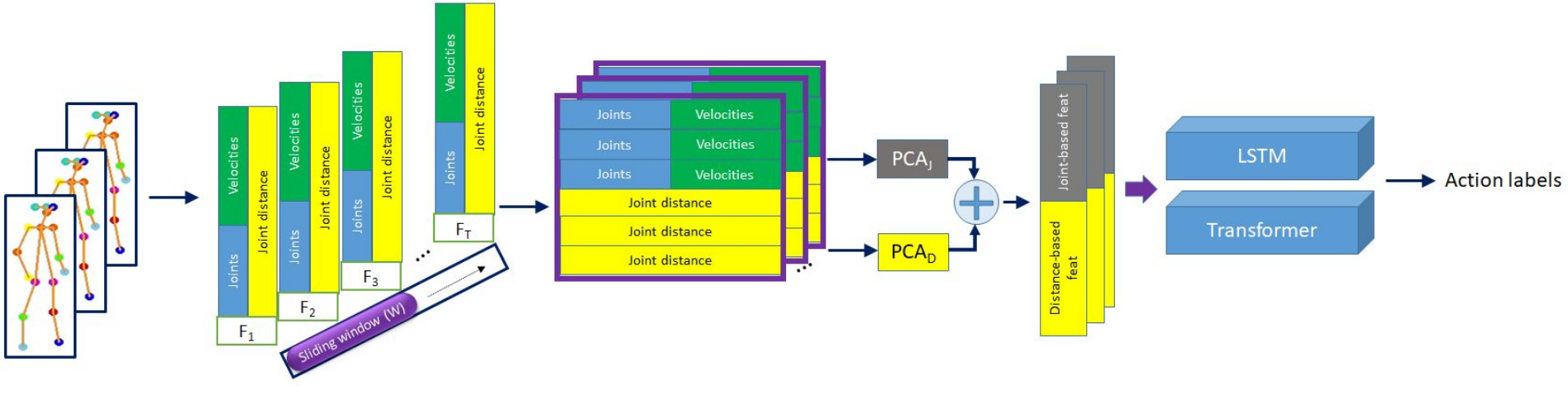
- **Advanced geometric features**
  - treat an action video as a sequence of action-snippets
    - temporal order -> overlapped combination of consecutive poses
    - effectively discriminates similar poses
  - highly discriminative representation of action-snippet
  - geometric relationship and body transition
  - less sensitive to the inter-class action variability

- **Adequate deep sequential neural networks(DSNN)**
  - BILSTM & Transformer
  - Concurrently model spatial relationships between geometric characteristics of different body parts
  - Capture the temporal dependencies in terms of inter-frame correlation

Methodology

- Architecture overview
- Preprocessing
- Data augmentation
- Feature extraction
- Deep Sequential Neural network models

Architecture overview



Preprocessing

- **OpenPose***
  - Convolutional Neural Network to produce two heatmaps
  - 18 or 25 joint coordinates
  - each joint -> a point with coordinate (x,y) in the 2D space

- **Dealing with missing data**
  - filled based on its relative position in the previous frame with respect to the neck

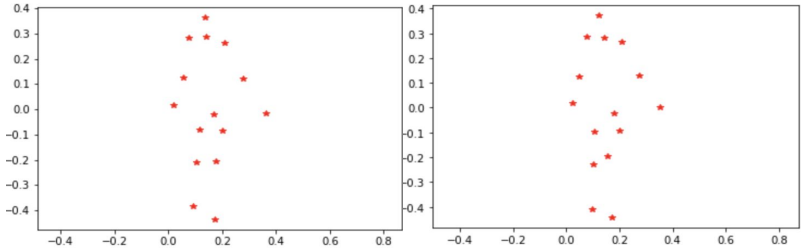
OpenPose* - Z. Cao, G. Hidalgo, T. Simon, S. -E. Wei and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 1, pp. 172-186, 1 Jan. 2021, doi: 10.1109/TPAMI.2019.2929257.

Data augmentation

1) Rotation (15, 30)

$$x_r = x\cos\theta - y\sin\theta$$

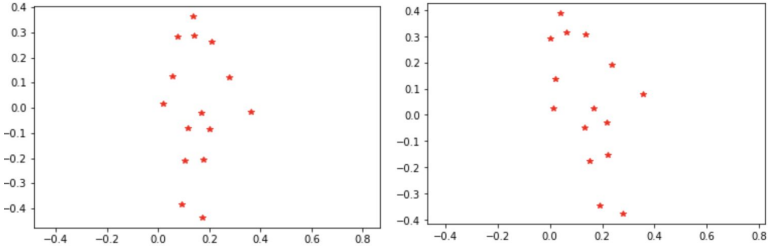
$$y_r = y\cos\theta + x\sin\theta$$



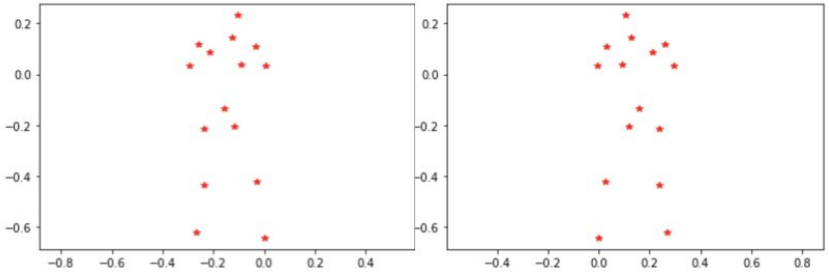
2) Pose-shifting

- Gaussian noise
  - σ = 0.002
  - σ = 0.004

$N(0,\ \sigma^2)$



3) Pose-flipping



Feature Extraction

- Sliding window - W
- Two kind of features:
  - Joint based
  - Distance-based
- Full set of skeletal joints:
  - $\mathcal{S} = \{\mathbf{s}_i^t = (x_i^t, y_i^t) | i \in [1, N], t \in [1, W]\}$
    - N - number of body joints
    - i-th joint $\mathbf{s}_i^t$ - point with 2D coordinate at frame t

Feature Extraction

Joint-based features

- A concatenated joint positions

$$\mathbf{s}_i = [s_i^1, s_i^2, ..., s_i^W]$$
$$\mathbf{s}_{joint} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N]$$

- An average height(H)
- The moving velocity of the body

$$\mathbf{v}_{\mathbf{body}} = \frac{\mathbf{v}_{\mathbf{center}}}{H}$$

$$\mathbf{v}_{\mathbf{center}} = [\mathbf{s}_0^1, (\mathbf{s}_0^2 - \mathbf{s}_0^1), ..., (\mathbf{s}_0^W - \mathbf{s}_0^{W-1})]$$

- The normalized joint positions(s)

$$\mathbf{s}_{norm,i}^t = \frac{\mathbf{s}_i^t - \bar{\mathbf{s}}_{\mathbf{joints}}}{H}$$

- The velocities of joints(v)

$$\mathbf{v}_i = [s_{n,i}^1, (s_{n,i}^2 - s_{n,i}^1), ..., (s_{n,i}^W - s_{n,i}^{W-1})]$$
$$\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N]$$

- Concatenated joint-based features

$$\mathbf{F}_{\mathbf{joint}} = [\mathbf{x}, \mathbf{v}_{\mathbf{body}}, \mathbf{v}]$$

Feature Extraction

Distance-based features

- **The Euclidean distance of each pairs (i, j)of joint positions**

$$d_{ij}^t = \sqrt{(x_i^t - x_j^t)^2 + (y_i^t - y_j^t)^2}$$

  - n(n−1)/2 distances in the frame t
  - Frame wise-representation

$$\mathbf{f}_{dist}^t = [d_1^t, d_2^t, ..., d_K^t]$$

  - Distance feature of an given snippet

$$\mathbf{F}_{dist} = [\mathbf{f}_{dist}^1, \mathbf{s}_{dist}^2, ..., \mathbf{s}_{dist}^W]$$

- **Concatenated reduced feature vectors**

$$\mathbf{F} = [\mathbf{F}_{\mathbf{joint}}, \mathbf{F}_{\mathbf{dist}}]$$

Deep Sequential Neural networks

BiLSTM

LSTM

- Retain or forget the information
- Determines how much information needs to be saved to current cell
- Controls information flow

$$F = (F^1, F^2 ... F^T)$$

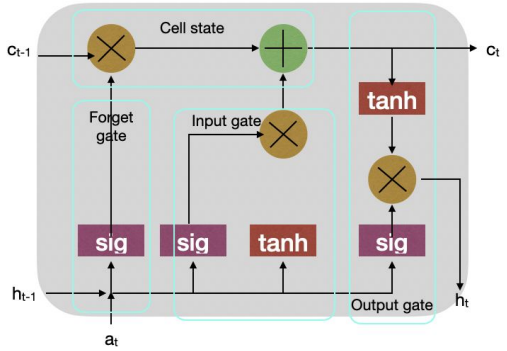
$$f_t = \sigma(U_f F_t + V_f h_{t-1} + b_f)$$

$$i_t = \sigma(U_i F_t + V_i h_{t-1} + b_i)$$

$$o_t = \sigma(U_o F_t + V_o h_{t-1} + b_o)$$

$$c_t = f_t * c_{t-1} + i_t * \tanh(\sigma(U_o F_t + V_o h_{t-1} + b_o))$$

$$h_t = o_t * \tanh(c_t)$$



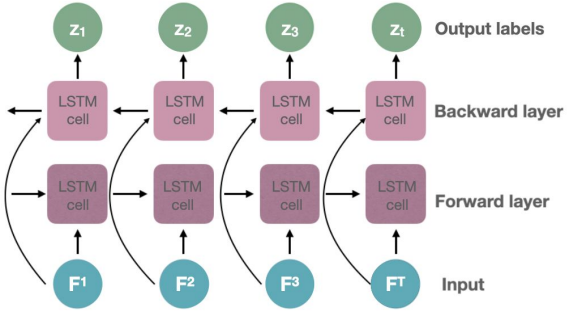
BiLSTM

- The first cell runs from past to future
- The second cell runs from future to past
- Backward and forward layer outputs are concatenated

$$\overrightarrow{h_t} = o_t * \tanh(c_t)$$

$$\overleftarrow{h_t} = o_t * \tanh(c_t)$$

$$z_t = [\overrightarrow{h_t}, \overleftarrow{h_t}].$$



Deep Sequential Neural networks

Transformer model

- Uses mechanism of self attention

$$Attention = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

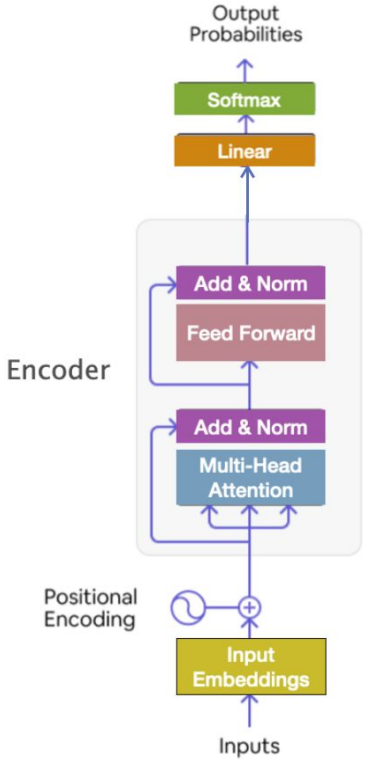
$$Q = FU^Q$$

$$K = FU^K$$

$$V = FU^V$$

  - Q - query matrix that is vector representation of one action-snippet in the video sequence
  - K - keys matrix that is vector representation of all action-snippets in video sequence
  - V - values of all action-snippets in vector representation

- Softmax function calculates the attention scores
- Self-attention layers - order agnostic
- Positional encoding embeds positions of action snippets



Experiments & Results

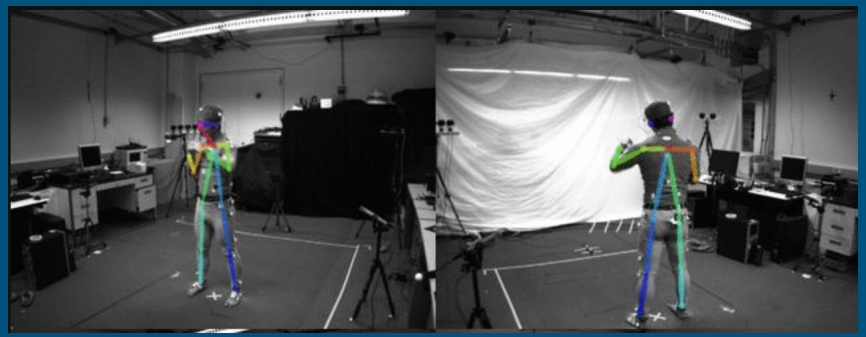
Dataset

JHMDB

- 928 videos
- 21 action types
- 32 frames
- 3 training and test splits

MHAD

- 1438 videos
- 6 action types
- 32 frames
- 18 joint positions



Window size effect on accuracy

- Sliding window size W=5
- JHMDB: 470 joint-based & 525 distance-based feature vector
- MHAD: 524 joint-based & 765 distance-based feature vector
- PCA algorithm reduces to 80 dimensions each vector

Dataset	Model	Window size	Accuracy
JHMDB	Transformer	1	60%
	Transformer	3	64%
	Transformer	5	**67%**
	Transformer	8	64%
JHMDB	BiLSTM	1	59%
	BiLSTM	3	63%
	BiLSTM	5	**66%**
	BiLSTM	8	62%
MHAD	Transformer	1	99%
	Transformer	3	97%
	Transformer	5	**99%**
	Transformer	8	98%
MHAD	BiLSTM	1	96%
	BiLSTM	3	98%
	BiLSTM	5	**99%**
	BiLSTM	8	97%

Experimental settings

Transformer

- Encoder layers (1,2,5)
- Number of heads(1,6,8, 16)
- Adam optimizer with decaying learning rate
- 30 epochs
- 64 batch size

Dataset	Batch sizes	Accuracy
JHMDB	16	70.9%
	32	71.5%
	64	**74.7%**
	128	71.8%
	256	70.7%
	512	68.7%

Dataset	Encoder layers	Number of heads	Accuracy without data augmentation	Accuracy with data augmentation
JHMDB	1	1	60.1%	66.8%
	1	6	61.3%	66.9%
	1	8	**62.4%**	**67.7%**
	1	16	**62.5%**	**70.9%**
	2	1	62.5%	65%
	2	6	60.3%	63%
	2	8	62.5%	64%
	5	1	60.2%	63.6%
	5	6	55.6%	59.7%
	5	8	55.7%	57.9%

Dataset	Encoder layers	Number of heads	Accuracy
MHAD	1	1	97.5%
	1	6	98.2%
	1	8	**99.3%**
	1	16	**99.4%**
	2	1	96.7%
	2	6	97.4%
	2	8	96.3%
	5	1	96.2%
	5	6	95.7%
	5	8	95.6%

Experimental settings

BiLSTM

- Hidden units(196)
- Batch size (2048)
- Adam optimizer with decaying learning rate
- 300 epochs

Dataset	Hidden units	Batch size	Decay rate	Accuracy
JHMDB	196	2048	0.96	70%
MHAD	196	2048	0.96	99%

Results and Complexity analysis

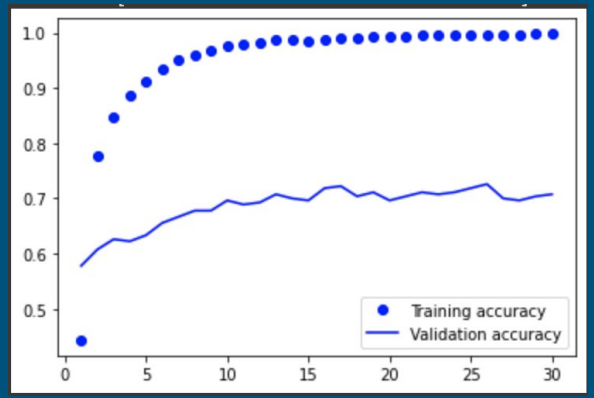
COMPARISON RESULTS ON JHMDB DATASET

Method	Parameters	Speed on GPU	Accuracy
DD-Net(filters=64)[31]	1.82M	2200 FPS	77.2%
DD-Net(filters=16)[31]	0.15M	3618 FPS	65.7%
DD-Net(with data augmentation)	1.82M	2200 FPS	74.4%
KNN	-	3450 FPS	59%
Random Forest	-	3745 FPS	63%
MLP	-	3750 FPS	61%
Transformer(layers=1, heads=16)	1.45M	3696 FPS	**74.7%**
Transformer(layers=1, heads=8)	0.83M	3875 FPS	**73.7%**
Transformer(layers=1, heads=1)	**0.11M**	3893 FPS	**69.7%**
Transformer(without data augmentation)	0.83M	3875 FPS	62%
BiLSTM	0.65M	3540 FPS	70%

COMPARISON RESULTS ON MHAD DATASET

Dataset	Methods	Accuracy
MHAD	LSTM&RNN[32]	97%
	KNN	95%
	Random Forest	96%
	MLP	97%
	BiLSTM	**99%**
	Transformer	**99%**

The learning curve of Transformer model on JHMDB dataset



Conclusion

Key contributions:

- Geometric representation and body transition properties of skeleton within action snippets capture the spatial and temporal relationship between poses

- Effective Transformer and BiLSTM architectures accurately learn the deep correlations of consecutive action-snippets in a long skeleton sequence

- Transformer model on JHMDB - 74.7%
- Transformer model on MHAD - 99%

Future work:

- Focus on generalization ability of the method
- Investigation for online action recognition

Thank you for your attention!

References:

[1] G. Paoletti, J. Cavazza, C. Beyan and A. Del Bue, "Subspace Clustering for Action Recognition with Covariance Representations and Temporal Pruning," 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 6035-6042, doi: 10.1109/ICPR48806.2021.9412060.

[2] L. Liu, S. Ma and Q. Fu, "Human action recognition based on locality constrained linear coding and two-dimensional spatial-temporal templates," 2017 Chinese Automation Congress (CAC), 2017, pp. 1879-1883, doi: 10.1109/CAC.2017.8243075.

[3] C. Huang, C. Hsieh, K. Lai and W. Huang, "Human Action Recognition Using Histogram of Oriented Gradient of Motion History Image," 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control, 2011, pp. 353-356, doi: 10.1109/IMCCC.2011.95.

[4] Mohana, Dr & U M, Mahanthesh. (2020). Human Action Recognition using STIP Techniques. International Journal of Innovative Technology and Exploring Engineering. 9. 10.35940/ijitee.G5482.059720.
[5]B. Liang and L. Zheng, "A Survey on Human Action Recognition Using Depth Sensors," 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2015, pp. 1-8, doi: 10.1109/DICTA.2015.7371223.

[6] J. Shan and S. Akella, "3D human action segmentation and recognition using pose kinetic energy," 2014 IEEE International Workshop on Advanced Robotics and its Social Impacts, 2014, pp. 69-75, doi: 10.1109/ARSO.2014.7020983.

[7] F. Angelini, Z. Fu, Y. Long, L. Shao and S. M. Naqvi, "2D Pose-Based Real-Time Human Action Recognition With Occlusion-Handling," in IEEE Transactions on Multimedia, vol. 22, no. 6, pp. 1433-1446, June 2020, doi: 10.1109/TMM.2019.2944745.

[8] D. Avola, M. Cascio, L. Cinque, G. L. Foresti, C. Massaroni and E. Rodolà, "2-D Skeleton-Based Action Recognition via Two-Branch Stacked LSTM-RNNs," in IEEE Transactions on Multimedia, vol. 22, no. 10, pp. 2481-2496, Oct. 2020, doi: 10.1109/TMM.2019.2960588.